UNITED STATES PATENT APPLICATION FOR:

SYSTEMS AND METHODS FOR NAVIGATING A GRAPHICAL HIERARCHY

Inventors:

Christopher E. Bales
Jeffrey Mueller
James Owen
Jalpesh Patadia
Nathan Olson
Manish Devgan
Timothy Noonan

CERTIFICATE OF MAILING BY "EXPRESS MAIL" UNDER 37 C.F.R. §1.10

"Express Mail" mailing label number: EV327622143US
Date of Mailing: 2/25/04

I hereby certify that this correspondence is being deposited with the United States Postal Service, utilizing the "Express Mail Post Office to Addressee" service addressed to: **MAIL STOP PATENT APPLICATION, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450** and mailed on the above Date of Mailing with the above "Express Mail" mailing label number.

Tina M. Galdos (Signature)
Name: Tina M. Galdos
Signature Date: 2/25/04

SYSTEMS AND METHODS FOR NAVIGATING A GRAPHICAL HIERARCHY

Inventors:

Christopher E. Bales
Jeffrey Mueller
James Owen
Jalpesh Patadia
Nathan Olson
Manish Devgan
Timothy Noonan

COPYRIGHT NOTICE

CLAIM OF PRIORITY

[0002] This application claims priority from the following application, which is hereby incorporated by reference in its entirety:

[0003] SYSTEMS AND METHODS FOR PORTAL AND WEB SERVER ADMINISTRATION, U.S. Application No. 60/451,174, Inventors: Christopher E. Bales, et al., filed on February 28, 2003. (Attorney's Docket No. BEAS-1371US0)

CROSS-REFERENCE TO RELATED APPLICATIONS

[0004] This application is related to the following co-pending applications which are hereby incorporated by reference in their entirety:

[0005] SYSTEMS AND METHODS FOR PORTAL AND WEB SERVER ADMINISTRATION, U.S. Application No. _____, Inventors: Christopher E. Bales, et al., filed on _____. (Attorney's Docket No. BEAS-1371US1)

[0006] SYSTEMS AND METHODS FOR CONTEXT-SENSITIVE EDITING, U.S. Application No. _____, Inventors: Christopher E. Bales, et al., filed on _____. (Attorney's Docket No. BEAS-1373US0)

[0007] SYSTEMS AND METHODS FOR AN EXTENSIBLE ADMINISTRATION TOOL, U.S. Application No. _____, Inventors: Richard Mousseau, filed on _____. (Attorney's Docket No. BEAS-1376US0)

[0008] SYSTEMS AND METHODS FOR PERSONALIZING A PORTAL, U.S. Application No. _____, Inventors: Christopher E. Bales, et al., filed on _____. (Attorney's Docket No. BEAS-1381US0)

[0009] CONTENT MINING FOR VIRTUAL CONTENT REPOSITORIES, U.S. Application No. _____, Inventors: Gregory Smith, et al., filed on _____. (Attorney's Docket No. BEAS-1483US0)

FIELD OF THE DISCLOSURE

[0010] The present invention disclosure relates to systems and methods for portal and web server administration.

BACKGROUND

[0011] Resources within web/application servers are many and varied (e.g., threads, servlets, roles, object pools, containers, etc.). However, conventional tools for performing administration and management of network accessible resources are often concerned with a gross level of detail. Such systems do not provide the types and extent of information desired by web/application server system administrators.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] **Figure 1** is an illustration of an administration system in an embodiment.

[0013] **Figure 2** is an illustration of a user interface that can be used to create and manage portal users and groups in an embodiment.

[0014] **Figure 3** is an illustration of a hierarchy browser zoom feature in an embodiment.

[0015] **Figure 4** is an illustration of a user interface that can be used to create and manage portals in an embodiment.

[0016] **Figure 5** is an illustration of a user interface that can be used to create and manage portal desktops in an embodiment.

[0017] **Figure 6** is an illustration of desktop resource/component hierarchy that has been created with a template.

[0018] **Figure 7** is an illustration of page layout context-sensitive editor in an embodiment.

[0019] **Figure 8** is an illustration of a user interface that can be used to create roles in an embodiment.

[0020] **Figure 9** is an illustration of a user interface that can be used to add groups to roles in an embodiment.

[0021] **Figure 10** is an illustration of a user interface that can be used to entitle a desktop in an embodiment.

[0022] **Figure 11** is an illustration of a user interface that can be used to entitle a page in an embodiment.

[0023] **Figure 12** presents two exemplary views of a user interface that can be used to manipulate a virtual content repository in one embodiment.

[0024] **Figure 13** is an illustration of a user interface that can be used to modify a user profile in an embodiment.

[0025] **Figure 14** is an illustration of a user interface that can be used to modify a placeholder definition in an embodiment.

[0026] **Figure 15** is an illustration of a user interface that can be used to create and modify user segment definitions in an embodiment.

[0027] **Figure 16** is an illustration of a user interface that can be used to create and modify properties associated with content in an embodiment.

[0028] **Figure 17** is an illustration of a user interface that can be used to create and edit content selectors in an embodiment.

[0029] **Figure 18** is an illustration of a user interface that can be used to create delegated administration roles in an embodiment.

DETAILED DESCRIPTION

[0030] The invention is illustrated by way of example and not by way of limitation in the figures of the accompanying drawings in which like references indicate similar elements. It should be noted that references to "an" or "one" embodiment in this

disclosure are not necessarily to the same embodiment, and such references mean at least one.

[0031] In one embodiment, a user interface provides a means for a user to interact with one or more processes that are operable to configure and manage portals and/or web servers. By way of a non-limiting example, a user interface can include one or more of the following: 1) a graphical user interface (GUI); 2) an ability to respond to sounds and/or voice commands; 3) an ability to respond to input from a remote control device (e.g., a cellular telephone, a personal digital assistant, or other suitable remote control); 4) an ability to respond to gestures (e.g., facial and otherwise); 5) an ability to respond to commands from a process on the same or another computing device; and 6) an ability to respond to input from a computer mouse and/or keyboard. This disclosure is not limited to any particular user interface. Those of skill in the art will recognize that many other user interface embodiments are possible and fully within the scope and spirit of this disclosure.

[0032] **Figure 1** is an illustration of an administration system in an embodiment. Although this diagram depicts objects/processes as logically separate, such depiction is merely for illustrative purposes. It will be apparent to those skilled in the art that the objects/processes portrayed in this figure can be arbitrarily combined or divided into separate software, firmware and/or hardware components. Furthermore, it will also be apparent to those skilled in the art that such objects/processes, regardless of how they are combined or divided, can execute on the same computing device or can be distributed among different computing devices connected by one or more networks or other suitable communication means.

[0033] In one embodiment and by way of a non-limiting example, the system can include a collection of administration user interfaces **100**, one or more web/application servers **102**, and one or more databases **104**, connected by one or more networks **106** or other suitable communication means. A network can include but is not limited to: public and/or private networks, wireless networks, optical networks, and satellite based communication links. Other suitable communication means can include but is not limited to: random access memory, file system(s), distributed objects, persistent storage, and inter-processor communication networks. The WebLogic® Server, available from BEA Systems, Inc., is a suitable web/application server in one embodiment. The one or

more databases can include but is not limited to: relational databases, object-oriented databases, file systems, or any other kind of persistent storage.

[0034] **Figure 2** is an illustration of a user interface that can be used to create and manage portal users and groups in an embodiment. By way of a non-limiting example, a user interface can be implemented using software such as X Windows or Microsoft® Windows. In one embodiment, a user interface can include two graphical components that can work together: an optional hierarchy browser **200** and a context-sensitive editor **202**. The hierarchy browser can render information such that hierarchical relationships between objects are apparent from the indentation of an object relative to other objects. For example, the object represented by the text "Everyone" is the root of the hierarchy. Its immediate children are "InternalUsers", "ExternalUsers" and "Employees". These children share "Everyone" as their parent. "InternalUsers" has one child, "MyInternalUser". Likewise, the parent of "MyInternalUser" is "InternalUsers". By way of a non-limiting example, selecting an object in the hierarchy browser can invoke a context-sensitive editor appropriate for editing the object. The rectangle **204** surrounding the object "MyInternalUser" indicates that this object has been selected. In one embodiment, objects can be manipulated (e.g., dragged and dropped) in the browser, thus allowing a user to easily rearrange the hierarchy.

[0035] In one embodiment, new groups can be created and nested within existing groups. For example, the global group "Everyone" has three nested groups: "Internal Users", "External Users" and "Employees". Although not shown in **Figure 2**, nested groups can also contain nested groups. Groups can be nested without limit. A group can contain zero or more nested groups and zero or more users. A user can belong to zero or more groups. Child groups are considered to fulfill the group membership of their parents. Although the present disclosure is not limited to any particular user interface or method of user interaction, in one embodiment new groups can be created within an existing group by right-clicking a mouse on the group (e.g., "Everyone", "InternalUsers", "ExternalUsers", or "Employees") and selecting an "Add Group" menu option (not shown). Users can be created in the same way by right-clicking a mouse on the group and selecting "Add New User". **Figure 2** illustrates a context sensitive editor for the user "MyInternalUser" which belongs to the group "InternalUsers". The name can be modified and password associated with this user can be modified via input fields **206** and **208**, respectively.

[0036] **Figure 3** is an illustration of a hierarchy browser zoom feature in an embodiment. In one embodiment, a "zoom" feature can be used to render a view of a hierarchy in a hierarchy browser with a root object other than the absolute root of the hierarchy. For example, the root of hierarchy **300** is the "Everyone" object. In one embodiment, a new object can be selected by the user such that the hierarchy is rendered with the new object as the root. For example, if "ExternalUsers" were chosen as the new root, the hierarchy can be rendered as in **302**. By zooming in, additional **304** objects can come into view that may have been clipped in the original rendering. In addition, the zoom feature can be used recursively such that a user can zoom in repeatedly. In one embodiment, a user can zoom in on an object by selecting it in some fashion (e.g., via a mouse click, a menu selection, keyboard input, etc.). A user can also "pop" or un-zoom zoomed views until the original view is visible. In one embodiment, popping a zoomed view can be accomplished through user interaction with the user interface (e.g., via a mouse click, a menu selection, keyboard input, etc.).

[0037] In addition to creating and managing users and user groups, the system can be used to create portals. A portal provides a way to aggregate content and integrate applications, allowing a visitor to a Web site to access everything via a user interface. Portals can be composed of a collection of portlets, each of which typically presents an application. Portlets are arranged on pages, which in turn are part of a book. These components are on the main body of the portal, which can also include a header and footer as part of the shell. The way that the portal is displayed and how it behaves is determined by the look and feel. It is this collection of components that makes up a portal. In one embodiment, a desktop can be a specific view of a portal, allowing for variations based on the characteristics of a visitor to a site. Thus, a desktop is a portal. For example, an employee and a customer might both visit a particular portal, but each can be directed to the appropriate desktop. Each desktop can have a distinct look and feel, organization of books and pages, and set of available portlets. Further control over the available resources is accomplished with Visitor Entitlements. The system allows users to create and configure all of these components.

[0038] **Figure 4** is an illustration of a user interface that can be used to create and manage portals in an embodiment. By way of non-limiting example, the hierarchal browser **400** presents a view of a portal resources or components tree. The "Portals" object represents the root of a portal subtree and is not itself a portal. In this example,

the "Portals" object has one child, a portal named "Portal 1". The rectangle **406** surrounding the "Portals" object indicates that it has been selected. Context-sensitive editor **402** allows a user to create and modify portals selected in the tree. Since a root portal object has been selected (and not a particular portal), the editor allows a new portal to be defined. Had "Portal 1" been selected, a user would be able to edit its properties. New portals can be created hierarchically below the "Portals" object at the level of the "Portal 1" object. After entering property information into the editor, the "Create New Portal" button **404** can be selected to create the portal. In this example, a new portal named "MyPortal" is being defined. After the button **404** has been selected, the new portal can appear in the portal subtree as shown in **Figure 5** below. The new portal can also include a textual description, uniform resource locator (URL) and a universal resource identifier (URI). It will be evident to those of skill in the art, and within the scope and spirit of this disclosure, that any properties serving any purpose can be associated with a portal.

[0039] **Figure 5** is an illustration of a user interface that can be used to create and manage portal desktops in an embodiment. Desktops provide user-specific views of portals. Desktops are related to each when they are part of the same portal, but they can be distinct in terms of their look and feel and their portal resources/components. Desktops provide an easy way for administrators to target specific users based on users' identity, group membership, or profile. Entitlements can be set on a desktop that allow them to be used by specific users or groups. By way of a non-limiting example, **Figure 5** shows that "My Portal" and "Portal 1" are direct children of the "Portals" object as shown in browser **500**. A desktop "Desktop 1" has been added to the portal "MyPortal". This can be accomplished any number of ways, including but not limited to right-clicking a mouse on "MyPortal" and selecting "Create new desktop" from a pop-up menu.

[0040] **Figure 5** illustrates that "Desktop 1" has been selected **504**. As such, context-sensitive editor **502** allows its properties to be changed. In one embodiment, properties can include: title, description and URL. It will be evident to those of skill in the art than many more properties can be associated with a desktop and still be within the scope and spirit of the present disclosure. Desktop resources can be automatically created for the desktop from a template. A template provides a way to create new desktops that are pre-configured with portal resources/components that are ready for use.

Templates are useful as the starting point for portal applications, allowing for the rapid deployment of new desktops in a portal. By selecting the "Create from Template" button **506**, a user can choose a template to apply to the desktop. In one embodiment, templates can include descriptive information to assist the user in making a selection.

[0041] **Figure 6** is an illustration of desktop resource/component hierarchy that has been created with a template. "Desktop 1" in optional hierarchy browser **600** has a book ("Main Page Book") which includes two pages ("Avitek Inweb", "My Page"). Each page includes a number of portlets. The "Avitek Inweb" page includes the following portlets which are represented graphically as children of the page: "Login", "News Feed" and "Portal Search". The "My Page" page includes the following portlets: "My Task List", "My Content", "Portal Search", "My Mail" and "My Contacts". To preview the new desktop in a browser, a user can right click Desktop1, and select "View Desktop" from a popup menu (not shown). When the portal is rendered (e.g., when viewed in a web browser), the book control corresponding to "Main Page Book" will allow the visitor to navigate to each page. When a page is rendered, its portlets will be rendered subject to entitlement restrictions, if any.

[0042] Portal resources can be associated with a desktop manually, rather than through the use of a template. In one embodiment and by way of a non-limiting illustration, resources from other parts of the resource hierarchy can be dragged and dropped (i.e. moved or copied) individually or in groups onto a desktop object in a hierarchy browser, thereby associating them with the desktop. Desktop resources can include portlets, books, pages, look and feels, shells, layouts, and other suitable resources. A page provides a way to organize portlets into groups based on related content, similar tasks, or simply user preference. Non-limiting examples include a human resources page that contains company-specific portlets, a finance page that includes banking and portfolio portlets, and a personal page that includes frequently accessed portlets. In one embodiment, a user can create a new page by selecting a book in a hierarchy browser and then selecting a "Create New Page" button in the corresponding context-sensitive editor. The new page will appear in the hierarchical view beneath the book. The page can then be selected and edited with the page context-sensitive editor. In addition, the book editor allows the book's pages to be ordered for presentation purposes at render time. By way of a non-limiting example, the following properties can be specified for a page: layout, locale, title, theme, portlets, entitlements and description.

[0043] **Figure 7** is an illustration of page layout context-sensitive editor in an embodiment. In one embodiment, a user can determine the overall layout of a portal page and the position of portlets on the page. The illustrated layout style in this example -- "three column layout" -- is indicated by pull-down menu **706**. The three columns are **700**, **702** and **704**. Other layouts are possible (e.g., two-column, grid, etc.) and are available from the pull-down menu. A user can dynamically switch between any layout, regardless of how the portlets are presently arranged, by selecting a layout from the menu. A rectangle representing the display area of "My Task List" portlet is displayed in column **700**. Column **702** contains portlets "My Content" and "Portal Search". Column **704** contains portlets "My Mail" and "My Contacts". Each rectangle representing the display area of a portlet can be resized larger or smaller to thereby increase or decrease its display area. In addition, the portlet rectangles can be moved between columns and rearranged within columns by dragging and dropping.

[0044] Visitor entitlements (or entitlements) can control access to portal application resources/components such as portlets, pages, and desktops. Entitlements can be set in a library or in portal applications. Entitlements can use roles and security policies to control access to resources. Roles dynamically group users based on username, group membership, profile, session and request attributes, and/or an assortment of date and time functions. Security policies determine what capabilities for a given resource are available to a given role. Entitlement capabilities can differ by resource and can include view, minimize, maximize, and edit.

[0045] **Figure 8** is an illustration of a user interface that can be used to create roles in an embodiment. A role can be used as part of an entitlement definition. Optional hierarchy browser **800** illustrates a "Visitor Entitlements" tree that has one immediate child, "Visitor Roles". Beneath "Visitor Roles", there are two roles defined: "Internal Users" and "External Users". When "Visitor Roles" is selected (as indicated by the rectangle **804**), editor pane **802** allows a user to create new roles by typing in a role name **806** and selecting a "Create New Role" button **808**. The new role is then added the hierarchy as illustrated in **Figure 9**.

[0046] **Figure 9** is an illustration of a user interface that can be used to add groups to roles in an embodiment. Optional hierarchy browser **900** a "Visitor Entitlements" tree that has one immediate child, "Visitor Roles". Beneath "Visitor Roles", there are three roles defined: "Internal Users", "External Users" and "Employee

Role". The role "Employee Role" is selected, as indicated by the surrounding rectangle **904**. Context-sensitive editor **902** illustrates a role editor corresponding to the selected role. User groups **906** can be added to the role definition by selecting the desired roles **910** and selecting the "Add to Role" button **908**.

[0047] **Figure 10** is an illustration of a user interface that can be used to entitle a desktop in an embodiment. Optional hierarchy browser **1000** illustrates a "Portal Resources" tree that includes a "Portals" subtree which in turn includes a portal "My Portal". The portal "My Portal" includes a desktop "Desktop 2" which is selected, as indicated by the rectangle **1004**. Context-sensitive editor **1002** allows modification of properties associated with "Desktop 2". In this example, the "Entitlements" tab **1006** has also been selected, therefore the context-sensitive editor for "Desktop 2" is tailored for editing entitlement information. A user can entitle roles with different capabilities for the selected desktop by selecting role(s) and associated properties (if any). By way of a non-limiting example, desktop capabilities can include the ability to view a desktop. Thus, visitors who belong to the groups embodied in the selected roles will be allowed to view the desktop. Here, visitors in the role of "Employee Role" will be able to view "Desktop 2" whereas those in "External Role" will not.

[0048] **Figure 11** is an illustration of a user interface that can be used to entitle a page in an embodiment. Optional hierarchy browser **1100** illustrates a "Portal Resources" tree that includes a "Library" subtree which in turn includes a "Pages" subtree. The "Pages" subtree includes two pages: "Avitek" and "My Page" which is selected, as indicated by the rectangle **1104**. Context-sensitive editor **1102** allows modification of properties associated with "My Page". In this example, the "Entitlements" tab **1106** has also been selected, therefore the context-sensitive editor for "My Page" is tailored for editing entitlement information. A user can entitle roles with different capabilities for the selected page by selecting the role(s) and associated properties (if any). By way of a non-limiting example, page capabilities can include the ability to view a page, the ability to edit information accessible through a page, the ability to rename page resources, and the ability to minimize or maximize portlet windows on the page. Here, visitors in the role of "Employee Role" have the ability to view, edit, rename, minimize and maximize. Whereas visitors in the "External Role" can only view the page. Other portal resources/components can be entitled in a similar fashion to desktops and pages.

[0049] Content management enables a user to integrate, manage, and personalize content in a portal environment. Content is a key component of any portal. Content can be defined as unstructured or semi-structured data. A common example is an image file and associated metadata; for example, date created, date modified, author, and subject. In Admin tool, a content type defines the shape of a content item. A content type can be any combination of binary, integer, calendar, string, Boolean, and properties. Interaction Management personalizes the delivery of content based upon these non-binary properties. Content can be organized into a content hierarchy. The top-level node is defined as a Virtual Content Repository. Under the Virtual Content Repository, you can plug in multiple, heterogeneous content repositories. This task is based on a single instance of the native BEA repository. Repositories can contain multiple hierarchy and content nodes. Hierarchy nodes function primarily as organizational units while content nodes function primarily as content items. Hierarchy nodes can be nested within each other infinitely. Content nodes are contained with hierarchy nodes and/or within the repository itself.

[0050] The system allows the content hierarchy to be reorganized and allows content properties to be edited. Users can add hierarchy nodes and content nodes. In one embodiment, nodes can be added in two different ways: through use of a batch loading utility or via a user interface. The following related U.S. patent application which is included herein in its entirety by reference includes information pertaining to batch loading a virtual content repository: CONTENT MINING FOR VIRTUAL CONTENT REPOSITORIES, U.S. Application No. _____, Inventors: Gregory Smith, et al., filed on _____.

[0051] **Figure 12** presents two exemplary views of a user interface that can be used to manipulate a virtual content repository in one embodiment. In one embodiment, there are hierarchy nodes **1204** and content nodes **1206**. A user can add either kind of node to the hierarchy by right-clicking on a node and selecting "Add Node" from a pop-up menu (not shown). Nodes can be moved by dragging and dropping them. Nodes can be renamed by right-clicking and selecting "Rename" from a pop-up menu (not shown). Nodes can deleted by right-clicking a node and selecting "Delete" from a pop-up menu (not shown). View **1200** shows a hierarchy before being manipulated. View **1202** shows the same hierarchy being manipulated: content nodes "a.jpg" and "b.jpg" were

dragged and dropped into the "External" hierarchy node; and content node "c.jpg" was dragged and dropped into "Internal" hierarchy node.

[0052] Personalization provides a way to deliver content to Web site visitors based upon various criteria. This includes information about the user (user profile), the users current session, the request made by the user, and other data. A personalized site provides the visitor with a better experience because the content displayed can be targeted to their interests. One way to deliver personalized content is via a Placeholder, which in one embodiment is comprised of a JSP tag and a definition. The JSP tag is used by a developer on a portlet JSP, and it refers to the Placeholder definition which contains the rules that determine which content to display. To tailor the content delivered in a Placeholder for specific users, a User Segment can be created and used in personalization definitions. User Segments provide dynamic classification of users based on various criteria. For applications where more than one content item is to be displayed, or where non-image data is to be displayed, Content Selectors are provided. These are similar to Placeholders in that they have a definition managed by the administrators. But Content Selectors differ from Placeholders in the way that a developer can use them.

[0053] **Figure 13** is an illustration of a user interface that can be used to modify a user profile in an embodiment. Optional hierarchy browser **1300** presents a view of a "User Groups" tree that has one immediate child (although there could be many more), the group "Internal Users". This group has one member, the group "Employees" which is currently selected **1304**. By selecting tab **1306**, a context-sensitive editor **1302** for the selected group is rendered. The editor allows the properties associated with a user (i.e., the user profile), to be modified. Although any property can be associated with a user, in this example there are two apparent: "Title" and "Type". Title currently has no value since its associated value is empty. Whereas the Type has a value of "Internal" **1308**. Properties can be added, deleted and their values changed by the editor.

[0054] **Figure 14** is an illustration of a user interface that can be used to modify a placeholder definition in an embodiment. Context-sensitive editor **1400** allows placeholder rules to be defined and edited for a given placeholder. Placeholder rules determine which content a placeholder will display on a portal page. A rule contains natural language phrases some of which are highlighted (e.g., in square brackets, underlined, etc.). A user can change highlighted phrases by selecting them. Highlighted phrases can be changed to new values that are appropriate for a given phrase's semantic

attributes. In this non-limiting example, the rule includes a preamble that states: "[All] of the following are [true]:". This means that all of the following phrases must be true in order for the rule to be evaluated to true. If the rule evaluates to true for a given content, then the content can be displayed by the placeholder. A user can change the highlighted phrase "All" to "Any" such that a logical OR is performed on the rule phrases rather than a logical AND. Likewise, "True" can be changed to "False" to reverse the logic of the rule.

[0055] A rule includes one or more phrases. In one embodiment, a rule phrase is in the form <property> <relationship> <value>, where <property> is a property defined on a virtual repository content node, <relationship> is a comparator (e.g., is less than, is greater than, is equal to, is not equal to, etc.) and <value> is the value of a given <property>. In this example, there is a single phrase: "[audience] [is equal to] [internal]". Each of these highlighted phrases can be changed by a user. The property phrase can be changed to be that of any property associated with content. Likewise, the value and relationship phrases can also be changed. Here, the audience property must be equal to "internal" in order for the rule to evaluate to true.

[0056] **Figure 15** is an illustration of a user interface that can be used to create and modify user segment definitions in an embodiment. Optional hierarchy browser **1500** can display a "User Segments" hierarchy. In this example, the segment "External" is selected, as indicated by the rectangle **1504**. Context-sensitive editor **1502** contains the user segment definition. The user segment definition can include natural language phrases some of which are highlighted (e.g., in square brackets, underlined, etc.). A user can change highlighted phrases by selecting them. Highlighted phrases can be changed to new values that are appropriate for a given phrase's semantic attributes. In this non-limiting example, the rule includes a preamble that states: "When [all] of these conditions apply:". This means that all of the following phrases must be true in order for the rule to be evaluated to true. If the rule evaluates to true for a given portal visitor, then the visitor is considered a member of the user segment. The highlighted phrase "All" can be changed to "Any" such that a logical OR is performed on the rule phrases rather than a logical AND. Likewise, "True" can be changed to "False" to reverse the logic of the rule.

[0057] A rule includes one or more rule phrases. In one embodiment, a rule phrase is in the form <property> <relationship> <value>, where <property> is a property

defined for a user profile, <relationship> is a comparator (e.g., is less than, is greater than, is equal to, is not equal to, etc.) and <value> is the value of a given <property>. In this example, there is a single phrase: "[type] [is equal to] [External]". Each of these highlighted phrases can be changed by a user. The property phrase can be changed to be that of any property associated with content. Likewise, the value and relationship phrases can also be changed. Here, the user type property must be equal to "External" in order for the rule to evaluate to true.

[0058] **Figure 16** is an illustration of a user interface that can be used to create and modify properties associated with content in an embodiment. Optional hierarchy browser **1600** displays a virtual content repository hierarchy that has a root of "VCR1". Beneath VCR1 is a repository "Ads". Ads contains hierarchy node "Financial", which contains hierarchy node "Ad Campaign", which contains hierarchy node "External". The External node contains content node "collegeplanning.jpg" which has been selected, as indicated by the rectangle **1604**. This node could be an image used to advertise college planning services. When selected, its properties become editable in context sensitive editor **1602**. The editor allows properties to be added and removed from the selected content node. In addition, property values can be changed. Here, the property "Audience" has a value of "External" which can be modified (e.g., by typing in field **1608**).

[0059] Dynamic personalization is fundamentally tied to the evaluation of rules based on a variety of properties. Content selectors can cause different content to be displayed in the a portlet based on dynamic evaluation of personalization rules. **Figure 17** is an illustration of a user interface that can be used to create and edit content selectors in an embodiment. Optional hierarchy browser **1700** can display a "Content Selectors" hierarchy. In this example, the content selector "tutorial" is selected, as indicated by the rectangle **1704**. Context-sensitive editor **1702** contains the rule definition of the selected content selector. The content selector definition can include natural language phrases some of which are highlighted (e.g., in square brackets, underlined, etc.). A user can change highlighted phrases by selecting them. Highlighted phrases can be changed to new values that are appropriate for a given phrase's semantic attributes. In this non-limiting example, the rule includes a preamble that states: "[All] of the following are [true]". This means that all of the following phrases must be true in order for the rule to be evaluated to true in order for content to be rendered in a portlet.

The highlighted phrase "All" can be changed to "Any" such that a logical OR is performed on the rule phrases rather than a logical AND. Likewise, "True" can be changed to "False" to reverse the logic of the rule.

[0060] A rule includes one or more rule phrases. In this example, there are two rule phrases. The first phrase is "[Audience] [is equal to] [External]". The rule requires that the audience property of any content to be displayed in a portlet that uses this content selector have its audience property equal to External. As discussed previously, each highlighted phrase can be selected and changed by the user. In addition, the user can add additional rule phrases. The second rule phrase is: "Visitor [is in any of the following user segments] [External, Internal]". This rule requires that a portal visitor belong to either the External or Internal user segment definitions. Each of these highlighted phrases can be changed by a user. The middle phrase can be changed to alter the relationship between Visitor and the value phrase. The value phrase can be selected to change the user segments (e.g., add segments and/or remove segments). Thus, this content selector will only display content that has the audience property equal to External and only when a portal visitor belongs to the External or Internal user segments.

[0061] The Delegated Administration feature facilitates localized administration of particular portal resources by designated portal administrators. For example, administration capabilities can be separately created and maintained for a company's Human Resources and Accounts Payable departments. The portal resources (e.g. content or user groups) associated with these departments can be managed by particular administrators who are specified and empowered via the use of the Admin tool.

[0062] **Figure 18** is an illustration of a user interface that can be used to create delegated administration roles in an embodiment. Optional hierarchy browser **1800** displays a delegated administration role hierarchy. There are two such roles in this example: "External" and "Internal". The External role is selected as indicated by the surrounding rectangle **1804**. Context-sensitive editor **1802** lists user groups **1806** available for inclusion in the selected delegated administration role. By selecting the check boxes adjacent to the user groups, said groups can be added to the definition of the delegated administration role. In addition, a user in a delegated administration role can further delegate management capabilities if the "Can Delegate" checkbox **1808** is selected.

[0063] In one embodiment, delegated administration roles can be empowered to manage portal resources/components (e.g., user profiles, group definitions, portals, desktops, pages, page layouts, roles, content repositories, placeholders, user segments and content selectors). Each of the prior resources/components can be provided with a user interface that allows delegated administration roles to be designated as having management capabilities over the given resource/component. Users who belong to a role can thus perform the management functions. Management capabilities vary depending on the resource, however, capabilities are generally include the ability to manage, create, read, update and/or delete. By way of a non-limiting example, if the resource is a content repository, the capabilities of a delegated administrator can include the ability to manage portions of the repository at and below specified hierarchy nodes.

[0064] One embodiment may be implemented using a conventional general purpose or a specialized digital computer or microprocessor(s) programmed according to the teachings of the present disclosure, as will be apparent to those skilled in the computer art. Appropriate software coding can readily be prepared by skilled programmers based on the teachings of the present disclosure, as will be apparent to those skilled in the software art. The invention may also be implemented by the preparation of integrated circuits or by interconnecting an appropriate network of conventional component circuits, as will be readily apparent to those skilled in the art.

[0065] One embodiment includes a computer program product which is a storage medium (media) having instructions stored thereon/in which can be used to program a computer to perform any of the features presented herein. The storage medium can include, but is not limited to, any type of disk including floppy disks, optical discs, DVD, CD-ROMs, microdrive, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, DRAMs, VRAMs, flash memory devices, magnetic or optical cards, nanosystems (including molecular memory ICs), or any type of media or device suitable for storing instructions and/or data.

[0066] Stored on any one of the computer readable medium (media), the present invention includes software for controlling both the hardware of the general purpose/specialized computer or microprocessor, and for enabling the computer or microprocessor to interact with a human user or other mechanism utilizing the results of the present invention. Such software may include, but is not limited to, device drivers, operating systems, execution environments/containers, and user applications.

[0067] The foregoing description of the preferred embodiments of the present invention has been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations will be apparent to the practitioner skilled in the art. Embodiments were chosen and described in order to best describe the principles of the invention and its practical application, thereby enabling others skilled in the art to understand the invention, the various embodiments and with various modifications that are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalents.